Money laundering detection using autoencoder graph neural networks

Author - Grama Tudor-Ionut
Babeș-Bolyai University

WeADL 2025 Workshop

The workshop is organized under the umbrella of WinDMiL, project funded by CCCDI-UEFISCDI, project number PN-IV-P7-7.1-PED-2024-0121, within PNCDI IV

uefiscdi
Unitatea Executivă pentru Finanțarea Învățământului Superior,

Outline

1. **Background and Fundamentals**
   - Money Laundering definition
   - Graph Neural Networks (GNN)
   - Autoencoders
2. **Literature review - GNN used in Money Laundering detection**
3. **Approach**
   - Original contributions
   - Methodology
   - Data representation - Graph
   - Dataset
   - GIN based AE architecture
   - PNA based AE adaptations
4. **Experimental Results**
   - Comparison to related work
   - Explainability
5. **Conclusion and future work**

Money Laundering definition

- Money Laundering is the process of making illicit funds hard or impossible to distinguish from those acquired by legal means,
- It is a complex process of obfuscation
- Financial institutions implement electronic anti money laundering systems, which function as an alert system that escalates messages that are further reviewed by data analysts
- Traditional approaches are typically rule-based system

Graph Neural Networks (GNN)

- The most common approach for developing graph neural networks is through the Message Passing Neural Network Framework.
- First, a message function aggregates the current states of neighboring nodes, as well as the features of connecting edges.
- Then a vertex update function updates the hidden state of a node v using the incoming messages from the neighboring nodes
- GNN message passing formalization:

$$m_{t+1}^{v} = \sum_{w \in N(v)} M_t(h_t^v, h_t^w, e_{vw}) \tag{1}$$

GNN - Message Passing

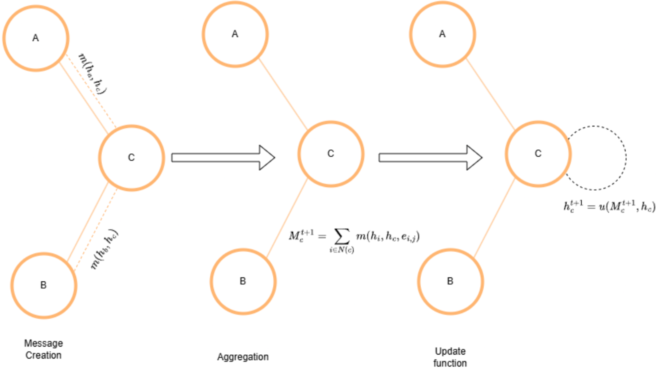

Figure: Process of message passing summarized [IGN24]. Created with `draw.io`

Autoencoders

- Autoencoders(AEs) are a class of neural networks
- Self-Supervised learning machine learning models
- The model learns how to reconstruct input data based on lower dimensional latent representations

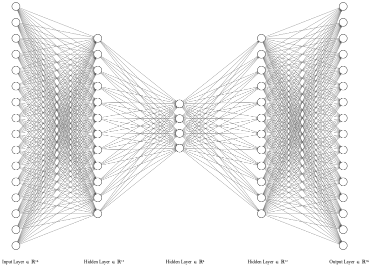

Figure: The general architecture of an autoencoder

Literature review - GNN used in Money Laundering detection

- Graph Neural Networks have been used successfully for anti money laundering tasks
- **Weber et al.** [WDC+19]
  - GCN with a skip connection for money laundering in bitcoin transactions
- **Johannessen and Jullum** [JJ23]
  - GNN architecture used to detect money laundering in a network of transaction from the largest Norwegian bank
- **Egressy et al.** [EvNB+24]
  - Combine key adaptations to standard graph neural networks in order to render them a better fit for graphs of transactions
- **Xu et al.** [XYW+24]
  - Rule Based systems combined with anomaly detection and graph neural network models

Original contributions

- A novel approach for detecting money laundering using autoencoder enhanced graph neural networks (GNNs)
- **Research questions**:
  - **[RQ1]** *Does integrating autoencoder components into the edge classification problem improve the predictive performance of money laundering detection?*
  - **[RQ2]** *What is the impact of integrating autoencoder components in a GNN-based architecture for the task of detecting illicit transactions in directed heterogeneous multigraphs?*
  - **[RQ3]** *What are the most significant features for detecting each type of transaction as provided by the SHAP explainability approach?*

Motivation

- We used this approach because we theorize that the nodes would be more inclined to preserve and encode features that are more important for defining the edges in their close proximity.
- Second, it helps differentiating edges with anomalous behavior, which is of interest, as we presume that fraudulent edges have some, albeit variable, specifics that ultimately make their representations deviate from expected patterns.

Methodology

- **Data representation**
  - Graphs of financial transactions
- **Building the GNN model**
  - Training the graph neural networks on the transaction graphs
- **Feature reconstruction**
  - The autoencoder encodes the node embeddings
    - Attempt to reconstruct edge features based upon node embeddings
    - The MSE given by the encoder is weighted by 0.1
- **Evaluation of results**
  - computing the F1 for the minority class

Data representation - Graph

Graph representation with nodes and edges

- Transactions are represented as edges
  - Edges are directed edges
  - Accounts are represented as nodes
  - Pairs of accounts can have multiple edges between them

Data representation - Graph

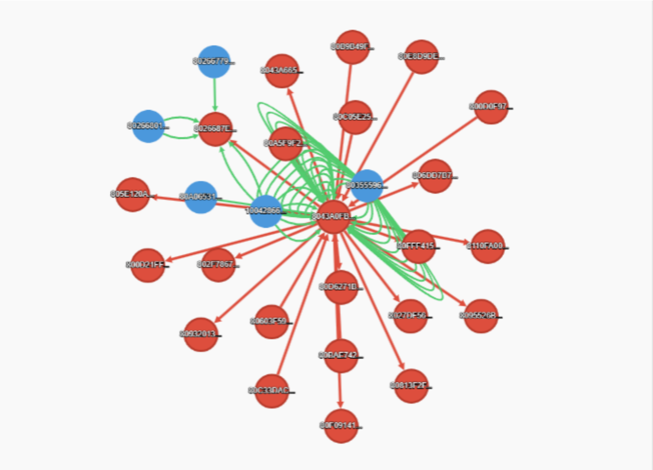

Figure: Example of financial data in graph format

Dataset

- Synthetic dataset created by IBM's AMLWorld model [AE+23]

Statistics	Higher Illicit	Lower Illicit
Number of Days Spanned	10	10
Number of Bank Accounts	515K	705K
Number of Transactions	5M	7M
Number of Laundering Transactions	3.6K	4.0K
Laundering Rate (1 per N Trans)	981	1942

Table: Statistics for the 'Small' subset of the dataset.

GIN based AE architecture

For both cases the experiments were conducted using the following parameters

- **Training Hyperparameters**
  - **Adam optimizer learning rate** 0.006
  - **Neighborhood Sampling:** [100, 100]
  - **Normal Transaction binary cross entropy weight:** 1.00
  - **Fraudulent Transaction binary cross entropy weight:** 7.1

GIN based AE architecture

- **Architecture of GIN based autoencoder:**
  - **Encoder:** Node embedding Linear ($num\_features \rightarrow 64$), Edge embedding Linear ($edge\_dim \rightarrow 64$), 2 GINEConv layers with residual connections and BatchNorm, Final encoder Linear ($64 \rightarrow 32$) to latent space
  - **Decoder:** Concatenate latent node pairs ($32 \times 2 = 64$), MLP ($64 \rightarrow 64 \rightarrow edge\_dim$) to reconstruct edge attributes
  - **Classifier:** Concatenate latent node features (64), original edge attributes ($edge\_dim$), and reconstruction error (1), then MLP ($65 + edge\_dim \rightarrow 50 \rightarrow 25 \rightarrow 2$) for final prediction

PNA based AE adaptations

- **PNA based autoencoder adaptations:**
  - **Reverse Message Passing:** Handle incoming and outgoing messages separately
  - **EgoIDs:** Flag which allows a node to detect whether it is part of a cycle
  - **Port Numbering:** Unique labels for each node
  - **Autoencoder:** The encoder and the decoder are now symmetric.

Comparison to related work

Table: Performance of baselines and proposed architectures.

Model	**AML Small HI**
LightGBM+GFs [AE+23]	62.86 ± 0.25
XGBoost+GFs [AE+23]	63.23 ± 0.17
GIN [XHLJ18, HLG+19])	28.70 ± 1.13
GIN+AE	34.98
PNA [VFH+18]	56.77 ± 2.41
GIN+EU [BH+18]	47.73 ± 7.56
R-GCN [SK+18]	41.78 ± 0.48
GIN+EgoIDs [YGSYL21]	39.65 ± 4.73
GIN+Ports [SYK19]	54.85 ± 0.89
GIN+ReverseMP [JN+19] +Ports	46.79 ± 4.97
GIN+Ports	56.85 ± 2.64
+EgoIDs (Multi-GIN)	57.15 ± 4.99
Multi-GIN+EU	64.79 ± 1.22
Multi-PNA	64.59 ± 3.60
Multi-PNA+EU	68.16 ± 2.65
Multi-PNA+EU+AE	50.92

Explainability

- SHAP values, with roots in game theory
- SHAP values computed using an XGBoost model
- Graph features converted to tabular format using the SNAPML library

Explainability

- Most important features are the payment format
- Features which refer to patterns are also very important
  - Such as the number of neighboring nodes, the number of transactions of a node, the ratios between them, etc.

Explainability

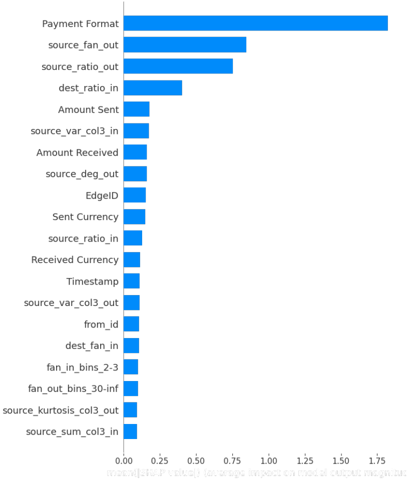

Figure: Global feature importance

Explainability

- Only the the fan/degree ratio at the source node make the transaction more likely to be fraudulent
- The transaction sent a large amount of money
- The transaction was simulated towards the beginning of the process.

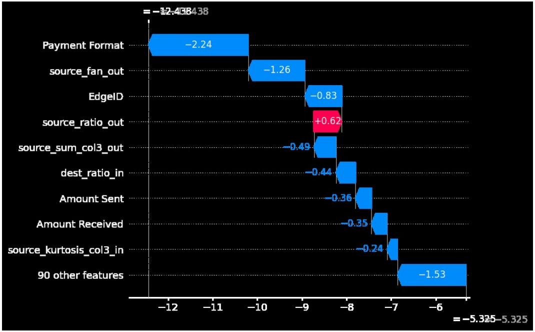

Figure: Normal transaction features

Explainability

- The transaction was conducted using the ACH format
- The transaction has a small ratio of fan/degree for its starting node
- The 5 most important features (those with col3) represent statistical features of the timestamp.

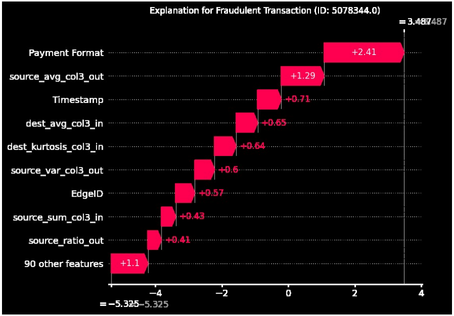

Figure: Fraudulent transaction features

Conclusion and future work

- **Conclusions**
  - We propose approaches for detecting money laundering in homogeneous and heterogeneous multigraphs using GNN enhanced with autoencoder components
  - Autoencoders enhance the classification performance for homogeneous graphs, but the performance decays for heterogeneous ones
  - Payment formats and the number of neighboring nodes of the participating accounts have a significant predictive capability in determining transactions involved in money laundering
- **Future work**
  - Implement more complex autoencoder components
  - Implement autoencoder components on other GNN architectures
  - Developing SHAP adaptations for explainability on edge labelling tasks, directly applicable to GNNs

Thank you!

Questions?

Bibliography I

Erik R. Altman, Béni Egressy, et al.
Realistic Synthetic Financial Transactions for Anti-Money Laundering Models.
In *Proceedings of NeurIPS 2023*, pages 1–24, 2023.

Peter W. Battaglia, Jessica B. Hamrick, et al.
Relational inductive biases, deep learning, and graph networks.
CoRR, abs/1806.01261:1–40, 2018.

Béni Egressy, Luc von Niederhäusern, Jovan Blanusa, Erik R. Altman, Roger Wattenhofer, and Kubilay Atasu.
Provably powerful graph neural networks for directed multigraphs.
In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on*

Bibliography II

Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada, pages 11838–11846. AAAI Press, 2024.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay S. Pande, and Jure Leskovec.
Pre-training graph neural networks.
CoRR, abs/1905.12265:1–22, 2019.

IGNNITION.
What is a gnn?, 2024.
Accessed: March 21, 2025.

Bibliography III

Fredrik Johannessen and Martin Jullum.
Finding money launderers using heterogeneous graph neural networks.
CoRR, abs/2307.13499:1–20, 2023.

Guillaume Jaume, An-phi Nguyen, et al.
edGNN: a Simple and Powerful GNN for Directed Labeled Graphs.
CoRR, abs/1904.08745:1–9, 2019.

Michael Sejr Schlichtkrull, Thomas N. Kipf, et al.
Modeling Relational Data with Graph Convolutional Networks.
In *Proceedings of ESWC 2018,*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer, 2018.

Bibliography IV

Ryoma Sato, Makoto Yamada, and Hisashi Kashima.
Approximation ratios of graph neural networks for combinatorial problems.
CoRR, abs/1905.10261:1–15, 2019.

Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm.
Deep graph infomax.
CoRR, abs/1809.10341, 2018.

Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I. Weidele, Claudio Bellei, Tom Robinson, and Charles E. Leiserson.
Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics.
CoRR, abs/1908.02591:1–7, 2019.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka.
How powerful are graph neural networks?
CoRR, abs/1810.00826:1–17, 2018.

Haosen Xu, Keke Yu, Ming Wei, Yida Zhu, and Yingchia Liu.
Intelligent anti-money laundering transaction pattern recognition system based on graph neural networks.
pages 1–9, December 2024.

Jiaxuan You, Jonathan M Gomes-Selman, Rex Ying, and Jure Leskovec.
Identity-aware graph neural networks.
Proceedings of the AAAI Conference on Artificial Intelligence, 35(12):10737–10745, May 2021.